Hi Caity!

Great job on the coding part - that was all spot on. For the discussion question, I would have liked to see you engage more with the readings. I think there are real questions not only of representation, but how this database is generated, structured, and maintained that would have a profound impact on your analysis.

A-

EDUC 423A/SOC 302A: Assignment 4

Caity McGinley

3/15/2021

# Honor Code Statement

We strongly encourage students to form study groups and students may discuss and work on assignments in groups. We expect that each student understands their own submission. As such, students must write their submissions independently and clearly disclose the names of all other students who were part of their study group. Additionally, lifting code or solutions directly from the internet (e.g., Google, GitHub, Stack Overflow) is a violation of the [Stanford Honor Code](https://communitystandards.stanford.edu/policies-and-guidance/honor-code). We take academic honesty and Honor Code violations extremely seriously and expect the same of students. If you have questions about what may or may not constitute an Honor Code violation, please reach out the teaching team.

**Email:** [caitym26@stanford.edu](mailto:caitym26@stanford.edu) **SUID:** 06211809 **Study Group:** Briefly talked to Sahi and Jim about Q 6 I acknowledge and agree to abide by the Honor Code.

**Signed:** Caity McGinley

**Question 1** Create a request URL that would return movies that have the word “data” in their title. Store it in an object called url.

**library**(rvest)

## Warning: package 'rvest' was built under R version 4.0.4

## Loading required package: xml2

**library**(RJSONIO)

## Warning: package 'RJSONIO' was built under R version 4.0.3

**library**(tidyverse)

## -- Attaching packages -------------------------------------------------------------------------------- tidyverse 1.3.0 --

## v ggplot2 3.3.2 v purrr 0.3.4  
## v tibble 3.0.3 v dplyr 1.0.2  
## v tidyr 1.1.2 v stringr 1.4.0  
## v readr 1.4.0 v forcats 0.5.0

## Warning: package 'readr' was built under R version 4.0.4

## Warning: package 'dplyr' was built under R version 4.0.3

## Warning: package 'stringr' was built under R version 4.0.3

## Warning: package 'forcats' was built under R version 4.0.3

## -- Conflicts ----------------------------------------------------------------------------------- tidyverse\_conflicts() --  
## x dplyr::filter() masks stats::filter()  
## x readr::guess\_encoding() masks rvest::guess\_encoding()  
## x dplyr::lag() masks stats::lag()  
## x purrr::pluck() masks rvest::pluck()

**library**(jsonlite)

## Warning: package 'jsonlite' was built under R version 4.0.4

##   
## Attaching package: 'jsonlite'

## The following object is masked from 'package:purrr':  
##   
## flatten

## The following objects are masked from 'package:RJSONIO':  
##   
## fromJSON, toJSON

**library**(tibble)  
**library**(purrr)  
**library**(tidyr)  
**library**(stringr)  
url = "http://www.omdbapi.com/?s=data&apikey=5c07a9aa"

**Question 2** Using the RSONIO package and the url you just created, request the data from OMDb.

OMDb <- **fromJSON**(url)

**Question 3** Show a tibble that includes the title, the year and imdbID, type and poster variables from your JSON object.

messy <- **as\_tibble**(OMDb)  
  
*#Results are nessed in own list*   
messy<- **map\_if**(OMDb, is.data.frame, list)   
  
hh <- **as\_tibble**(messy)   
  
*#unnesting dataframe*   
unnested <- **unnest**(hh)

## Warning: `cols` is now required when using unnest().  
## Please use `cols = c(Search)`

unnested <- unnested **%>%**   
 **mutate**(Year = **str\_remove\_all**(Year, pattern = "[[:punct:]]")) **%>%**   
 **mutate**(Year = **str\_replace\_all**(Year, "20062008", "2006, 2008")) **%>%**   
 **select** (**-** Response, **-** totalResults)  
   
 myData <- unnested[**-c**(9), ]  
   
myData

## # A tibble: 9 x 5  
## Title Year imdbID Type Poster   
## <chr> <chr> <chr> <chr> <chr>   
## 1 Data Tutashkhia 1977 tt01827~ seri~ https://m.media-amazon.com/imag~  
## 2 Olsenbanden og Data-H~ 1978 tt01323~ movie https://m.media-amazon.com/imag~  
## 3 Digimon Data Squad 2006, ~ tt11383~ seri~ https://m.media-amazon.com/imag~  
## 4 Platinum Data 2013 tt22829~ movie https://m.media-amazon.com/imag~  
## 5 RDG: Red Data Girl 2013 tt26229~ seri~ https://m.media-amazon.com/imag~  
## 6 Data Science Pioneers~ 2019 tt11023~ movie https://m.media-amazon.com/imag~  
## 7 The Human Face of Big~ 2014 tt33121~ movie https://m.media-amazon.com/imag~  
## 8 Data Limite segundo C~ 2014 tt39679~ movie https://m.media-amazon.com/imag~  
## 9 The Joy of Data 2016 tt59094~ movie https://m.media-amazon.com/imag~

**Question 4** Change the request URL so it only returns movies.

movieurl <- "http://www.omdbapi.com/?s=data&t=movie&apikey=5c07a9aa"  
  
movies <- **fromJSON**(movieurl)

**Question 5** Pick a movie you’d like to know more about and write a new URL to obtain a tibble that includes a description of the movie. Extract the plot and the names associated to the movie.

*#Digimon sounds interesting*   
  
digimon <- "http://www.omdbapi.com/?t=Digimon+Data+Squad&plot=full&apikey=5c07a9aa"  
  
digimovie <- **fromJSON**(digimon)  
  
digi <- **as\_tibble**(digimovie)  
  
digi <- digi **%>%**   
 **select**(Title, Actors, Plot)  
  
digi

## # A tibble: 1 x 3  
## Title Actors Plot   
## <chr> <chr> <chr>   
## 1 Digimon Da~ Quinton Flynn, Colleen O'Shaug~ "We pick up with our heroes and t~

**Question 6** Imagine that we would scale this up and scrape all movies with “data” in the title (note that the omdapi only returns the first 10 movies. You could get additional pages by tweaking the page parameter). You’d be interested in exploring how often the word “data” occurs in movie titles over time and what characteristics are associated with these movies (e.g., language, country, writers, actors, awards, ratings) because you are trying to get a better understanding of the mainstream depiction of data science. Using the W8 readings, write a critical discussion of using the OMDB API for such a research endeavor.

Using OMDB API for such a research endeavor doesn’t come with much harmful implications but could be a tedious process as a researcher. While web scraping can be “ugly” (Pittard), the data from OMDb came in a clean, interpretable json format that needed only a bit of unnesting. With the R package, it became easily readable. In addition, the website provided an API key free of charge for the first 1,000 scrapes. To get a better depiction of the mainstream presentation of “data science” changing the page parameter to include more movies would be a must. However, the website shows you how to change the link to change the parameters of what is scraped—it’s really simple. It gets a bit tricker when trying to analyze the depiction of data science based off the movies available. The data feminism readings encourages data scientists to recognize how data is a product of unequal social relations. For movie this can be seen in how the field is mainly white people, with white males having a majority of the roles and the lines. Data science movies might show more white males in positions of power. It’s important to recognize that this may or may not mirror the real life circumstances revolving the composition of data scientists. Data has the power to expose inequalities! Pointing out over representation of certain groups is needed as well as providing insight on why it might be like that is an expectation I would have of researcher looking at OMDb data. Yes - the data is easy to access, but are there features related to the construction of OMDb that might be important for your analysis?